Academic FAQ Chatbot using Retrieval Augmented Generation

Anung B Ariwibowo[a1], Syandra Sari[a2], Is Mardianto[b3], Gagah P Bangsa[b4], Chaerudin Saputra[a5], Jofi T Hakim[a6], M Ichsan Gunawan[a7]

[a]Information Systems Department, Universitas Trisakti, Indonesia
[b]Informatics Engineering Department, Universitas Trisakti, Indonesia
e-mail: [1]anung@trisakti.ac.id, [2]syandra_sari@trisakti.ac.id, [3]mardianto@trisakti.ac.id, [4]064002100036@std.trisakti.ac.id, [5]065002200023@std.trisakti.ac.id, [6]065002200023@std.trisakti.ac.id, [7]m.ichsan@trisakti.ac.id

Abstrak

Metode tradisional untuk mengakses informasi dari dokumen pedoman akademik sering kali menyebabkan pencarian manual yang tidak efisien dan pertanyaan berulang. Penelitian ini mengatasi tantangan tersebut dengan mengembangkan chatbot akademik Pertanyaan yang Sering Diajukan (FAQ) menggunakan pendekatan Retrieval Augmented Generation (RAG). Studi ini bertujuan untuk menyederhanakan akses informasi akademik pada dokumen akademik melalui chatbot berbasis Kecerdasan Artifisial. Metodologi mencakup pengumpulan dokumen, pra-pemrosesan data, dan pembuatan representasi vektor melalui teknik embedding. Sistem retrieval dibangun menggunakan basis data vektor untuk pencarian berbasis kemiripan, yang kemudian diintegrasikan dengan Model Bahasa Besar (LLM)Penelitian ini menggunakan model All-MiniLM-L6-v2 untuk embedding, sementara model LLM Gemini-2.0-flash digunakan untuk membangkitkan respons. Antarmuka chatbot dikembangkan menggunakan framework Flask. Survei pengguna menunjukkan kepuasan yang lebih tinggi terhadap hatbot berbasis RAG yang dikembangkan dibandingkan dengan layanan tradisional. Penelitian ini berkontribusi secara signifikan dengan mendemonstrasikan solusi AI yang efektif, adaptif, dan hemat biaya yang meningkatkan layanan informasi akademik.

Kata kunci: *Chatbot, Retrieval Augmented Generation, Model Bahasa Besar, Informasi Akademik*

Abstract

Traditional methods of accessing information from academic guideline documents often lead to inefficient manual searches and repetitive queries. This research addresses these challenges by developing an academic Frequently Asked Question (FAQ) chatbot using the Retrieval Augmented Generation (RAG) approach. The study aims to simplify academic information access on academic documents through an AI-integrated chatbot. The methodology includes document collection, data preprocessing, and vector representation via embedding techniques. A retrieval system is built using a vector database for similarity-based searches, integrated with a Large Language Model (LLM). This research used All-MiniLM-L6-v2 model for embeddings, while Gemini-2.0-flash is utilized as the LLM for response generation. The chatbot interface is developed using Flask framework. User surveys indicate higher satisfaction with the developed RAG-based chatbot compared to traditional services. This research contributes significantly by demonstrating an effective, adaptive, and cost-efficient AI solution that enhances academic information services.

Keywords: *Chatbot, Retrieval Augmented Generation, Large Language Model, Academic Information*

1. Introduction

The rapid advancements in artificial intelligence (AI) have instigated a significant digital transformation across various sectors, including education. A prominent innovation in this technological landscape is the emergence of Large Language Models (LLMs), which possess advanced capabilities in understanding, processing, and generating natural and contextual text [1]. Chatbots, widely recognized applications of LLM technology, enable users to interact by

writing textual prompts and receiving comprehensive responses. Meanwhile in academic settings, there is a growing demand for rapid, accurate, and personalized information access to support students, lecturers, and researchers in their learning and research activities.

Despite the potential of chatbots, conventional systems often exhibit limitations in comprehending complex contexts and delivering adequate responses. Academic institutions frequently face the challenge of recurring questions posed to non-teaching staff, necessitating prompt replies. Furthermore, access to educational and teaching information often relies on manual processes, such as downloading documents from faculty websites, which can be time-consuming and inefficient for finding specific details. Students commonly resort to manual keyword searches or seeking information from peers, leading to inefficiencies, especially when dealing with extensive and complex documents, and potentially hindering their understanding of academic policies. An efficient and responsive information system is therefore crucial to serve as a central reference for the entire academic community.

To address these challenges, leveraging LLMs for developing academic chatbots presents a promising solution to enhance the efficiency and quality of information services in educational institutions. A particularly effective approach is the integration of Retrieval-Augmented Generation (RAG). RAG [2], [3] combines the strengths of large language models with the ability to retrieve information from external knowledge bases. This method enables chatbots to deliver responses that are not only accurate but also highly relevant, grounded in reliable sources. Within a university context, such knowledge sources can encompass research journals, lecture modules, academic guidelines, and institutional repositories. By utilizing RAG, chatbots can extract specific information from these documents and generate contextual, in-depth answers, thereby improving the quality of user interaction and ensuring that provided information is trustworthy and consistent with academic standards. This makes chatbots an effective virtual assistant for accessing academic information, answering complex questions, and more efficiently supporting learning and research processes. In general, universities possess numerous academic documents, including research implementation guidelines, academic technical instructions, and Standard Operating Procedures (SOPs). Large Language Model and RAG technology can help facilitate the academic community's access to relevant information.

This research specifically focuses on the development of an academic chatbot for academic documents. The primary objective of this research is to develop a RAG-based chatbot, integrated with Artificial Intelligence (AI), to simplify access to academic information related to education and teaching within the university. This study addresses the following key problems:

1. How to implement the LLM with the Retrieval-Augmented Generation (RAG) approach to develop an academic chatbot capable of retrieving information from internal organization academic documents?
2. How effective is the LLM and RAG-based chatbot in providing accurate and relevant responses based on the research guidelines, academic technical instructions, and SOPs owned by organization?
3. What technical and non-technical challenges are encountered in integrating LLM and RAG?

2. Research Method / Proposed Method

The development of the Academic Chatbot prototype starts with conducting first survey to gain understanding of what students need to fulfill their academic information. Based on this survey, appointed academic documents were selected to be processed further in the form of embedding representation. A prototype of web-based application was developed with help from Python-based Flask framework [4], and finally the early prototype is used by students and lecturers. They assess the functionality of the chatbot to fulfill the academic information needs.

Quantitative method is used for analyzing the success of the Academic Chatbot prototype, carried out by surveying user respondents. The survey is conducted in two stages: the first survey before development to understand the main problems students face in obtaining academic information, and the second one was conducted after development to measure user satisfaction. Information collected from the survey includes user satisfaction with chatbot responses, chatbot speed in generating responses, and overall system response time. Users can be students, lecturers, or academic staff. A total of 36 students served as respondents in the initial survey conducted prior to the development phase of the chatbot. This initial survey was performed to assess the need and main issues students faced in obtaining academic information.

The RAG approach combines the power of large language models with the ability to retrieve information from external knowledge bases, enabling the chatbot to provide accurate and relevant responses based on trusted sources. RAG generally consists of three main steps: Indexing, Retrieval, and Generation.

3. Literature Study

ChatGPT has garnered worldwide attention since its launch in November 2020. Several years later, its creator, OpenAI, developed large language models with a significantly greater number of parameters [1]. ChatGPT is a technology based on the Generative Pre-trained Transformer (GPT) architecture [5], in which the underlying language model has undergone a pre-training process within a machine learning context. GPT technology itself is built upon the Transformer architecture [6], which has been the catalyst for the remarkable advancements in Artificial Intelligence witnessed in recent years.

Various GPT-based chatbots are now available, aside from ChatGPT, namely Gemini AI from Google[7], Microsoft Copilot from Microsoft, and Llama from Meta [8], among others. It is important to note that building these large language models (LLMs) requires immense computational resources [9], [10], [11]. This requirement is one of the key drawbacks of LLMs, particularly when it comes to answering questions that are highly specific to a particular problem domain.

Researchers, academics, and the public have implemented a variety of creative applications for Large Language Models (LLMs). To apply an LLM to a specific knowledge domain, two primary approaches exist: fine-tuning and Retrieval Augmented Generation (RAG). Fine-tuning is a further "learning" stage applied to a pre-trained LLM [12]. The main challenge with fine-tuning is the immense computational resources it demands, requiring computers with very large RAM and high-end CPUs [12], [13], [14]. For many organizations, this represents a significant barrier.

An alternative approach is Retrieval-Augmented Generation (RAG). This technique does not require the extensive computational resources needed for fine-tuning because it relies on a retrieval process [2], [3], [15]. Simply put, RAG uses an initial prompt from the user to search for relevant information from an organization's external data sources. This retrieved information is then used as a subsequent prompt to generate a final response. The steps involved in the RAG process are illustrated in Figure 1.

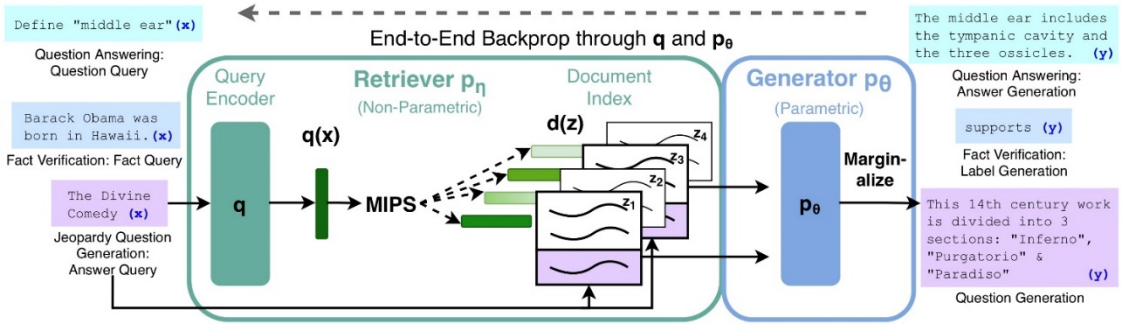

Figure 1. Conceptual working of Retrieval-Augmented Generation (RAG) [3]

Information within a document collection is converted into vector embeddings [16], [17], [18], [19]. This vector embedding representation is fundamentally how LLMs generate responses from user prompts. Nevertheless, the RAG technique also has its drawbacks, such as a slower response generation time compared to fine-tuning. Additionally, RAG requires an extra component to store the Document Index shown in Figure 1. This component is a vector database, which holds the embedding representations of the information from the document collection [15].

4. Result and Discussion

The development process of Academic Chatbot begins with selecting academic documents for initial development. In this study, we selected the Technical Guidelines for Academic Activities and New Student Admission Guide. These two documents were considered

sufficiently representative for the initial development of this Academic Chatbot. These documents were then going through cleaning and extracting process into a uniform plain text format. Within the chatbot application, technical guideline documents will be indexed by an administrator user where she may upload selected documents deemed relevant to the academic information need of the chatbot users. Behind the scenes the logic of chatbot application will segment the texts found in the uploaded documents into smaller, digestible pieces (*chunks*) to accommodate the context limitations of language models.

These chunks are then encoded into vector representations using an embedding model. The model used in the chatbot application is sentence-transformers/all-MiniLM-L6-v2. Before using that model, we tried to use all-mpnet-base-v2 as well. Although it shows good performance, `all-MiniLM-L6-v2` is chosen for its efficiency and accuracy, being smaller and faster for our final implementation. After chunking, the indexing and embedding results are stored in a vector database, namely FAISS (Facebook AI Similarity Search) [4], [20], [21].

Upon receiving a user query, the RAG system uses the same encoding model (used during the indexing phase) to transform the query into a vector representation. The system calculates the similarity score between the query vector and the chunk vectors in the indexed corpus. This vector representation of the chunked information from uploaded documents can be used as semantic search, like finding similarities between the terms "libur" and "cuti". The system prioritizes and retrieves the top-K chunks that show the greatest similarity with the query. These chunks are then used as extended context in the prompt. Interaction with reference documents is used to retrieve information.

In the generation step, which is the third step, user's question and the selected documents are synthesized into a coherent prompt. This research uses Google Gemini-2.0-flash to help formulate the final response. Gemini is chosen for its ability to understand various data types (text, images, PDFs, videos), better readability, more positive responses, and the availability of a free API compared to the paid ChatGPT API. The model's approach to answering can vary based on specific task criteria, allowing it to leverage inherent parametric knowledge or restrict its responses to the information contained in the provided documents. Any existing conversation history can be integrated into the prompt, enabling the model to engage in multi-turn dialogue effectively.

In developing our chatbot application, several supporting technologies and tools are used. For web application framework, Flask [4] is used to accelerate the web-based application of the chatbot. As for LLM, an open-source framework designed to facilitate the development of LLM-powered applications, LangChain [22], [23], is used to connect LLMs with various external data sources. To extract text from academic documents in PDF format, we used PyPDF2 library. RecursiveCharacterTextSplitter is used to break down long texts into smaller chunks.

5. Conclusion

This research focuses solely on the development of a RAG-based chatbot with the primary focus on accessing academic information in the context of higher education institutions, using textual data only. The business process is focused on managing academic information mainly for students who need to access academic information, while at the same time reducing the allocated academic personnel to answer those needs. This research only covers the development and initial testing phases of the chatbot, and early assessment of how well it served students. It does not cover long-term maintenance or updates.

Future research and development for this academic chatbot can significantly expand its capabilities and impact. Several ideas and key areas for future exploration include the following: Integration with existing institution’s information systems, enhanced user interface and user experience, robust knowledge management and dynamic document updates, exploration of additional LLMs, and enhancing current prototype with multi modal capabilities using multi modal models.

It is also our plan to analyze questions posed by early users of this version of chatbot, to modify more specific needs of academic information by students and lecturers as well.

6. Acknowledgement

The authors would like to express their gratitude to Notebook LM [24] for providing valuable help in various stages of this research. Notebook LM assists in brainstorming ideas in the early stages of development of this chatbot prototype, writing refinement and language

enhancement for the clarity of the manuscript. It is important to note that the authors reviewed, verified, and validated Notebook LM's responses to ensure accuracy, reliability, and compliance to academic standards.

References

[1] I. Solaiman dkk., "OpenAI Report Release Strategies and the Social Impacts of Language Models," 2019.
[2] Y. Gao dkk., "Retrieval-Augmented Generation for Large Language Models: A Survey," Des 2023, [Daring]. Tersedia pada: http://arxiv.org/abs/2312.10997
[3] P. Lewis dkk., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Mei 2020, [Daring]. Tersedia pada: http://arxiv.org/abs/2005.11401
[4] M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2 ed. O'Reilly Media, 2018.
[5] G. Yenduri dkk., "Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions," Mei 2023, [Daring]. Tersedia pada: http://arxiv.org/abs/2305.10435
[6] A. Vaswani dkk., "Attention Is All You Need," Jun 2017, [Daring]. Tersedia pada: http://arxiv.org/abs/1706.03762
[7] M. Imran dan N. Almusharraf, "Google Gemini as a next generation AI educational tool: a review of emerging educational technology," 1 Desember 2024, Springer. doi: 10.1186/s40561-024-00310-z.
[8] H. Touvron dkk., "LLaMA: Open and Efficient Foundation Language Models," Feb 2023, [Daring]. Tersedia pada: http://arxiv.org/abs/2302.13971
[9] M. U. Hadi dkk., "Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects," 16 November 2023. doi: 10.36227/techrxiv.23589741.
[10] H. Naveed dkk., "A Comprehensive Overview of Large Language Models," Jul 2023, [Daring]. Tersedia pada: http://arxiv.org/abs/2307.06435
[11] W. X. Zhao dkk., "A Survey of Large Language Models," Mar 2023, [Daring]. Tersedia pada: http://arxiv.org/abs/2303.18223
[12] V. B. Parthasarathy, A. Zafar, A. Khan, dan A. Shahid, "The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities," Agu 2024, [Daring]. Tersedia pada: http://arxiv.org/abs/2408.13296
[13] C. Jeong, "Fine-tuning and Utilization Methods of Domain-specific LLMs."
[14] A. Singh, N. Pandey, A. Shirgaonkar, P. Manoj, dan V. Aski, "A Study of Optimizations for Fine-tuning Large Language Models," Jun 2024, [Daring]. Tersedia pada: http://arxiv.org/abs/2406.02290
[15] R. C. Barron dkk., "Domain-Specific Retrieval-Augmented Generation Using Vector Stores, Knowledge Graphs, and Tensor Factorization," Okt 2024, [Daring]. Tersedia pada: http://arxiv.org/abs/2410.02721
[16] R. Egger, "Text Representations and Word Embeddings: Vectorizing Textual Data," dalam Tourism on the Verge, vol. Part F1051, Springer Nature, 2022, hlm. 335–361. doi: 10.1007/978-3-030-88389-8_16.
[17] W. L. Hamilton, J. Leskovec, dan D. Jurafsky, "Diachronie word embeddings reveal statistical laws of semantic change," 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers, vol. 3, hlm. 1489–1501, 2016.
[18] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, dan W. Wang, "Language-agnostic BERT Sentence Embedding," Jul 2020, [Daring]. Tersedia pada: http://arxiv.org/abs/2007.01852
[19] Y. Goldberg dan O. Levy, "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method," no. 2, hlm. 1–5, Feb 2014, Diakses: 15 Juli 2014. [Daring]. Tersedia pada: http://arxiv.org/abs/1402.3722
[20] "Welcome to Faiss Documentation — Faiss documentation." Diakses: 4 Agustus 2025. [Daring]. Tersedia pada: https://faiss.ai/index.html
[21] "Welcome to Flask — Flask Documentation (3.1.x)." Diakses: 4 Agustus 2025. [Daring]. Tersedia pada: https://flask.palletsprojects.com/en/stable/
[22] M. Irfan, "Penerapan Retrieval Augmented Generation Menggunakan Langchain dalam," 2024.

[23] "Build a Retrieval Augmented Generation (RAG) App: Part 1 | 🦜🔗 LangChain." Diakses: 4 Agustus 2025. [Daring]. Tersedia pada: https://python.langchain.com/docs/tutorials/rag/
[24] E. Tufino, "NotebookLM: An LLM with RAG for active learning and collaborative tutoring," Jul 2025, [Daring]. Tersedia pada: http://arxiv.org/abs/2504.09720